IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising the operations of:

analyzing an API of an application to determine a plurality of methods comprising the API;

exercising the API utilizing a test program that calls methods from the plurality of methods comprising the API;

generating a list of methods called by the test program; and

comparing the list of methods called by the test program with the plurality of methods comprising the API; and

determining a percentage of the plurality of methods comprising the API that are called by the test program and designating the application as portable when the percentage is greater than a predetermined bar percentage.

2.-3. (cancelled)

4. (Currently amended) A method as recited in claim 1 [[3]], further comprising the operation of designating the application as non-portable when the percentage is less than the predetermined bar percentage.

5. (Currently amended) A method ~~as recited in claim 1,~~ for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising the operations of:

analyzing an API of an application to determine a plurality of methods comprising the API;

exercising the API utilizing a test program that calls methods from the plurality of methods comprising the API;

generating a list of methods called by the test program;

comparing the list of methods called by the test program with the plurality of methods comprising the API;

~~further comprising the operation of~~ intercepting method calls from the test program utilizing stubs for corresponding methods of the plurality of methods comprising the API; and

recording the intercepted method calls and passing the intercepted method calls to corresponding methods of the plurality of methods comprising the API.

6.-7. (Cancelled)

8. (Currently amended) A computer program embodied on a computer readable medium for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising:

a code segment that analyzes an API for an application to determine a plurality of methods comprising the API;

a code segment that intercepts method calls utilizing stubs for corresponding methods of the plurality of methods comprising the API;

a code segment that generates a list of intercepted methods calls; and

a code segment that compares the list of intercepted methods calls with the plurality of methods comprising the API, wherein the method calls are generated by a test program that exercises the application API by calling methods from the plurality of methods comprising the API; and

a code segment that determines a percentage of the plurality of methods comprising the API that are called by the test program, wherein the application is designated as portable when the percentage is greater than a predetermined bar percentage.

9.-11. (Cancelled)

12. (Currently Amended) A computer program as recited in claim 8 [[11]], wherein the application is designated as non-portable when the percentage is less than the predetermined bar percentage.

13. (Currently amended) A computer program ~~as recited in claim 8,~~ embodied on a computer readable medium for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising:

a code segment that analyzes an API for an application to determine a plurality of methods comprising the API;

a code segment that intercepts method calls utilizing stubs for corresponding methods of the plurality of methods comprising the API;

a code segment that generates list of intercepted methods calls;

a code segment that compares the list of intercepted methods calls with the plurality of methods comprising the API; and

~~further comprising~~ a code segment that passes the intercepted method calls to corresponding methods of the plurality of methods comprising the API.

14. (Currently amended) A computer program ~~as recited in claim 8,~~ embodied on a computer readable medium for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising:

a code segment that analyzes an API for an application to determine a plurality of methods comprising the API;

a code segment that intercepts method calls utilizing stubs for corresponding methods of the plurality of methods comprising the API;

a code segment that generates list of intercepted methods calls;

a code segment that compares the list of intercepted methods calls with the plurality of methods comprising the API; and

~~further comprising~~ a code segment that reads a deployment descriptor for the application to determine a public interface for the application.

15. (Original) A computer program as recited in claim 14, further comprising a code segment that introspects class Java code for the public interface to generate a list of the plurality of methods comprising the API.

16. (Currently amended) A system for ascertaining application program interface (API) coverage for a Java 2 platform Enterprise Edition (J2EE) application, comprising:

logic that analyzes an API for an application to determine a plurality of methods comprising the API;

logic that generates a list of methods called by a test program; ~~and~~

logic that compares the list of methods called by the test program with the plurality of methods comprising the API to determine a percentage of the plurality of methods comprising the API that are called by the test program; and

logic that designates the application as portable when the percentage is greater than a predetermined bar percentage.

17. (Original) A system as recited in claim 16, wherein the test program exercises the application API by calling methods from the plurality of methods comprising the API.

18. (Cancelled)

19. (Original) A system as recited in claim 16, further comprising logic that designates the application as non-portable when the percentage is less than the predetermined bar percentage.

20. (Original) A system as recited in claim 19, further comprising logic that designates the application as non-portable when an exception occurs during execution of the test program.